Amendments to the Specification

The reference to Figure 37 in the Brief Description of the Drawings portion of the Specification was inadvertently omitted. Thus, please replace the section entitled Brief Description of the Drawings of the original specification with the following new section.

BRIEF DESCRIPTION OF THE DRAWINGS

--The same reference numbers are used throughout the figures to reference like components and features.

Fig. 1 is a graphical representation of a conventional filter graph representing a user-defined development project.

Fig. 2 is a block diagram of a computing system incorporating the teachings of the described embodiment.

Fig. 3 is a block diagram of an example software architecture incorporating the teachings of the described embodiment.

Fig. 4 is a graphical illustration of an example software-enabled matrix switch, according to an exemplary embodiment.

Fig. 5 is a graphical representation of a data structure comprising a programming grid to selectively couple one or more of a scalable plurality of input pins to a scalable plurality of output pins of the matrix switch filter, in accordance with one aspect of the described embodiment.

Fig. 6 is a graphical illustration denoting shared buffer memory between filters, according to one aspect of the described embodiment.

Fig. 7 is a flow chart of an example method for generating a filter graph, in accordance with one aspect of the described embodiment.

Fig. 8 is a flow chart of an example method for negotiating buffer requirements between at least two adjacent filters, according to one aspect of the described embodiment.

Fig. 9 graphically illustrates an overview of a process that takes a user-defined editing project and composites a data structure that can be used to program the matrix switch.

Fig. 10 graphically illustrates the project of Fig. 9 in greater detail.

Fig. 11 shows an exemplary matrix switch dynamically generated in support of the project developed in Figs. 9 and 10, according to one described embodiment.

Fig. 12 illustrates a graphic representation of an exemplary data structure that represents the project of Fig. 10, according to one described embodiment.

Figs. 13-18 graphically illustrate various states of a matrix switch programming grid at select points in processing the project of Figs. 9 and 10 through the matrix switch, in accordance with one described embodiment.

Fig. 19 is a flow chart of an example method for processing media content, in accordance with one described embodiment.

Fig. 20 illustrates an example project with a transition and an effect, in accordance with one described embodiment.

Fig. 21 shows an exemplary data structure in the form of a hierarchical tree that represents the project of Fig. 20.

Figs. 22 and 23 graphically illustrate an example matrix switch programming grid associated with the project of Fig. 20 at select points in time, according to one described embodiment.

Fig. 24 shows an example matrix switch dynamically generated and configured as the grid of Figs. 22 and 23 was being processed, in accordance with one described embodiment.

Fig. 25 shows an exemplary project in accordance with one described embodiment.

Fig. 26 graphically illustrates an example audio editing project, according to one described embodiment.

Fig. 27 depicts an example matrix switch programming grid associated with the project of Fig. 26.

Fig. 28 shows an example matrix switch dynamically generated and configured in accordance with the programming grid of Fig. 27 to perform the project of Fig. 26, according to one described embodiment.

Fig. 29 illustrates an exemplary media processing project incorporating another media processing project as a composite, according to yet another described embodiment.

Fig. 30 graphically illustrates an example data structure in the form of a hierarchical tree structure that represents the project of Fig. 29.

Figs 31-37 [[6]] graphically illustrate various matrix switch programming grid states at select points in generating and configuring the matrix switch to implement the media processing of Fig. 29.

Fig. 38 illustrates an example matrix switch suitable for use in the media processing project of Fig. 29, according to one described embodiment.

Fig. 38a graphically illustrates an example data structure in the form of a hierarchical tree structure that represents a project that is useful in understanding composites in accordance with the described embodiments.

Fig. 39 is a flow diagram that describes steps in a method in accordance with one described embodiment.

Fig. 40 illustrates an example method of generating a filter graph, in accordance with one aspect of the present invention.

Fig. 41 graphically illustrates an example reuse list, according to one aspect of the present invention.

Fig. 42 illustrates an example method for source combining in support of the method introduced in Fig. 40, according to one embodiment of the present invention.

Fig. 43 graphically illustrates a timeline representation of source combining introduced in Fig. 42.

Fig. 44 illustrates a block diagram of an example render engine utilizing a segment filter in a filter graph, in accordance with one aspect of the present invention.

Fig. 45 illustrates a flow chart of an example method of generating a filter graph to reuse source filters, in accordance with one aspect of the present invention.

Fig. 46 illustrates a flow chart of an example method of executing a development project utilizing a segment filter, in accordance with one aspect of the present invention.--

In the Action, the Office requests that the status of the copending applications at pages 1 and 8-10 be updated. Thus, please replace the section entitled Related Application on page 1 of the original specification with the following new section.

--**RELATED APPLICATION**

This application is a divisional application of and claims priority to U.S. Patent Application Serial No. 09/731,490, now U.S. Patent No. 6,983,466, the disclosure of which is incorporated by reference herein.--

Also, please replace the section entitled Related Applications on pages 8-11 of the original specification with the following new section.

--**Related Applications**

This application is related to the following commonly-filed U.S. Patent Applications, all of which are commonly assigned to Microsoft Corp., the disclosures of which are incorporated by reference herein:

- Application Serial No. 09/731,560, now U.S. Patent No. 6,774,919, entitled "An Interface and Related Methods for Reducing Source Accesses in a Development System", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/732,084, now U.S. Patent No. 6,834,390, entitled "A System and Related Interfaces Supporting the Processing of Media Content", naming Daniel J. Miller and Eric H. Rudolph as inventors;

- Application Serial No. 09/732,452, now U.S. Patent No.7,114,161, entitled "A System and Related Methods for Reducing Memory Requirements of a Media Processing System", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/731,529, now U.S. Patent No. 6,961,943, entitled "A System and Related Methods for Reducing the Instances of Source Files in a Filter Graph", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/732,087, now U.S. Patent No. 6,959,438, entitled "An Interface and Related Methods for Dynamically Generating a Filter Graph in a Development System", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/732,090, now U.S. Patent No. 6,611,215, entitled "A System and Related Methods for Processing Audio Content in a Filter Graph", naming Daniel J. Miller and Eric H. Rudolph as inventors, and now U.S. Patent No. 6,611,215;
- Application Serial No. 09/732,085, now U.S. Patent No. 7,114,162, entitled "A System and Methods for Generating an Managing Filter Strings in a Filter Graph", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/731,491, now U.S. Patent No. 6,768,499, entitled "Methods and Systems for Processing Media Content", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/731,563, now U.S. Patent No. 6,954,581, entitled "Systems for Managing Multiple Inputs and Methods and Systems for Processing Media Content ", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/731,892, now U.S. Patent No. 6,912,717, entitled "Methods and Systems for Implementing Dynamic

Properties on Objects that Support Only Static Properties", naming Daniel J. Miller and David Maymudes as inventors;

- Application Serial No. 09/732,089, now U.S. Patent No. 7,103,677, entitled "Methods and Systems for Efficiently Processing Compressed and Uncompressed Media Content", naming Daniel J. Miller and Eric H. Rudolph as inventors;
- Application Serial No. 09/731,581, now U.S. Patent No. 7,287,226, entitled "Methods and Systems for Effecting Video Transitions Represented By Bitmaps", naming Daniel J. Miller and David Maymudes as inventors;
- Application Serial No. 09/732,372, now U.S. Patent No. 6,882,891, entitled "Methods and Systems for Mixing Digital Audio Signals", naming Eric H. Rudolph as inventor; and
- Application Serial No. 09/732,086, now U.S. Patent Publication No. 2002/0099789 A1, entitled "Methods and Systems for Processing Multi-media Editing Projects", naming Eric H. Rudolph as inventor.

Various described embodiments concern an application program interface associated with a development system. According to one example implementation, the interface is exposed to a media processing application to enable a user to dynamically generate complex media processing tasks, e.g., editing projects. In the discussion herein, aspects of the invention are developed within the general context of computer-executable instructions, such as program modules, being executed by one or more conventional computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, personal digital assistants, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. In a distributed computer environment, program modules may be located in both local and remote memory storage devices. It is noted, however, that modification to the architecture and methods described herein may well be made without deviating from spirit and scope of the present invention. Moreover, although developed within the context of a media processing system paradigm, those skilled in the art will appreciate, from the discussion to follow, that the application program interface may well be applied to other development system implementations. Thus, the media processing system described below is but one illustrative implementation of a broader inventive concept.--